Memory Hierarchies and Caches

1) (10 points) Assume a 32-bit machine with a 128 Kbyte cache (total, but not including tags) and 32-byte blocks. Assume the cache is direct-mapped. For the address 0x1400fa24, what are the tag, index, and offset? Please write your answers in binary.

2) (10 points) Assume you have a cache with exactly 8 frames and that there is one word per frame. Compare direct-mapped vs. fully-associative for the following stream of accesses: 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, ... (repeating sequence of 9 addresses). What lesson do you learn from this experiment? Would a victim cache help?

3) (10 points) When might it make sense for a cache hierarchy NOT to enforce inclusion? (Inclusion means that any block in the L1 must be in the L2.) Hint: think about the possible relative sizes of the two caches.

4) (10 points) H&P 5.13a (only part a!)

Exploring Caches with SimpleScalar

5) (40 points) Start with the sim-cache simulator and just use the gcc and go benchmarks. You will NOT have to modify the sim-cache.c code for this assignment, but you will have to feed it different command line parameters to configure it. If you run sim-cache without any input parameters, it will spit out all of the possibilities, which should help you to figure out how to specify the configurations in the following experiments. Keep the default configuration for everything, unless instructed otherwise in the following experiments.

Experiment #1 explores the tradeoff between associativity and hit latency for a 4 KByte L1 data cache: Compare the performances of (a) 1-cycle direct-mapped cache, (b) 2-cycle 2-way set-associative cache, and (c) 3-cycle 4-way set-associative cache. Do not modify the other default characteristics of the L1 D$ (e.g., block size, total cache size, replacement policy). However, note that sim-cache makes you specify the number of sets—make sure that the caches you specify are all 4 KB in total size (not including tags)!

Experiment #2 explores the importance of caching for out-of-order processors. Using sim-R10K and just the benchmark anagram, compare a 1K vs. 4K direct-mapped L1 data cache for both in-order and out-of-order processors (assume the default configuration of sim-R10K for all other parameters besides L1 data cache and in-order vs. out-of-order).

Analysis: Given what you learned from these 2 experiments, (a) explain how you might choose to trade-off associativity vs. hit latency, and (b) explain what the results in experiment #2 say about the relative importance of the L1 D-cache for in-order vs. out-of-order processors.